AUTOMATING THE UMP OPEN REGISTRATION (OR) SYSTEM FOR COURSE TIMETABLING PROBLEM

KANGESWARY A/P KUPPUSAMY

THESIS REPORT SUBMITTED IN FULFILMENT OF THE DEGREE OF BACHELOR OF COMPUTER SCIENCE
(COMPUTER SYSTEMS & NETWORKING)

FACULTY OF COMPUTER SYSTEMS & SOFTWARE ENGINEERING

2013

Abstract

The open registration (OR) system is different from the "Program-based Registration System" which allows student to register their academic subject based on their own study plan. UMP's students are required to make the combination of the study plan to make sure their timetable is suitable with their favorite time. There are some methods on solving the timetabling such as Tabu Search (Qu et al.., 2009), Hill Climbing (Appleby et al 2011), Simulated Annealing (Kirkpatrick and Vecci, 1983) and Great Deluge Algorithm (Dueck,1993) which had been used to solve the college or University timetabling problems. These searching methods fulfilled the automated timetabling system and applied on many systems. These researches will benefit the UMP's student to view the course information with more specific details and convenience the process course registration online.

Abstrak

Sistem pendaftaran terbuka (OR) adalah berbeza daripada "Sistem Pendaftaran berasaskan Program" yang membolehkan pelajar mendaftar subjek akademik mereka berdasarkan pelan kajian mereka sendiri. Pelajar UMP adalah diwajibkan untuk membuat gabungan pelan kajian untuk memastikan jadual mereka sesuai dengan masa kegemaran mereka. Terdapat beberapa kaedah kepada penyelesaian penjadualan waktu seperti Tabu Search (Qu et al .. 2009), Hill Climbing (Appleby et al 2011), simulasi Penyepuhlindapan (Kirkpatrick dan Vecci, 1983) dan Great Deluge Algoritma (Dueck, 1993) yang telah digunakan untuk menyelesaikan kolej atau masalah penjadualan University. Kaedah-kaedah ini mencari dipenuhi sistem penjadualan automatik dan digunakan pada banyak sistem. Penyelidikan ini akan memberi manfaat kepada pelajar UMP untuk melihat maklumat kursus dengan butir-butir yang lebih khusus dan kemudahan pendaftaran kursus proses dalam talian.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In Semester one, 2011/2012, University Malaysia Pahang develops an Open Registration System that is OR System to University Malaysia Pahang student's for register courses by online. Previously University Malaysia Pahang developed "Program-about Registration System" where a student study plans are pre-determined by the faculty based on the Program Course Structure. The Open Registration System is an easier menu system that allows student to register their subject based on their own timetabling.

An order the University Malaysia Pahang Academic Initializations, students can register their course up to 19 credits hours for each semester. Students have to plan their own timetable because the 12 credits is the lowest credit to register subject. Therefore those who want to register up to 19 credits per semester also can. It allows student to graduate early of the semester and the same time can save their time and money.

Nowadays University Malaysia Pahang student using course catalog to arrange their subject based on timetable. Course catalog content lists of all courses are provided on the certain semester along with their time and location. In Open Registration, most of the subjects are offered in every year. Instead of, the categories of each offered course in normally half of categories of

the student enrolled. For example, if Data Network and Security course (BCN 2023) is a requirement for all Bachelor of Computer Science third year's students with the total 280 students only.

1.1 Problem Statement

Now in the registration system, as students have to manually check their classing and type of courses that available in course catalog. The course catalog included the course, the time and classes. There have many combination of timetable that and really confuses went the students was selecting the most appropriate timetable. In this system, does not appear time went the classes is clashed. However, UMP proposes a student course registration system that able to identify the correct timeslot and classes. As well as calculate the cost value. Objectives and the scopes will present in the next section.

1.2 Objectives

The objective of the system is:

- To develop a prototype of UMP courses registration system allows students to register courses based on section and timetable.
- To develop a system we able to give suggestion of the courses that available in timetable.
- To implement the course time tabling by using solve concept from the soft ump constraints as guides for student satisfaction.

1.3 Scopes

The scope of this system is:

- Only degree students are able to register subject and drop the registered subject.
- Only the courses from faculty FSKKP and university subject available to register.
- The module only include courses from semester I 2011/12 and semester II 2011/12.

1.4 Thesis Organization

In thesis organization include with five chapters. The chapter I will discuss on introduction to system, problem statement of the system, objective and scope of the system. In chapter 2 I will discuss about literature review by using some of the references to do research and describe about the existing problem or solution done by other parties. For the references, we can get from book, journal, conference, proceeding, magazine, thesis or website. Next about chapter 3, in this chapter I will describe about methodology system and also overall approach and framework of research. After that about chapter 4, I will discuss on design on how to convert the development process into the form of a document and also about the project design will be explained also. In Chapter 5, I will conclude what I have done through in the all system.

CHAPTER 2

LITERATURE REVIEW

This chapter briefly describes the review on existing techniques related with attendance system. This chapter comprises four sections: The first section explained more briefly about University timetabling problem. In the second section, describes details on problem statement that applied to solve the timetabling problem. The third section describes about method that use to create timetabling. The fourth section explained on the MATLAB. Fifth section explained on techniques that will be used in this system. Finally, sixth section describes the previous existing system.

2.1 Overview of Open Registration

The open registration (OR) system is different from the "Program-based Registration System" which allows student to register their academic subject based on their own study plan. UMP's students are required to make the combination of the study plan to make sure their timetable is suitable with their favorite time. There are some methods on solving the timetabling such as *Tabu Search* (Qu et al.., 2009), *Hill Climbing* (Appleby et al 2011), *Simulated Annealing* (Kirkpatrick and Vecci, 1983) and *Great Deluge Algorithm* (Dueck,1993) which had been used to solve the college or University timetabling problems. These searching methods fulfilled the automated timetabling system and applied on many systems. These researches will benefit the UMP's student to view the course information with more specific details and convenience the process course registration online.

2.2 UMP Course Timetabling Problem

University timetabling problem is very hard to solve for optimally [3]. It is very difficult to solve with traditional methods and the amount of computation to find optimal solution increase exponentially due to the large enrolment of students every years [9]. The university course timetabling problem involves assigning a set of courses, student and lecturers to a specific number of rooms and timeslots [1] with the weekly assignment of a set of lectures [13]. It is difficult to manage timetable went, there are no time and which subject clash on that time. The constraints that have to be satisfied by a timetable are usually divided into two categories there are hard constraints and soft ones. Hard constraints are those constraints that must be rigidly fulfilled. The following diagram shows an overview of open subject registration which is the reference from [12].

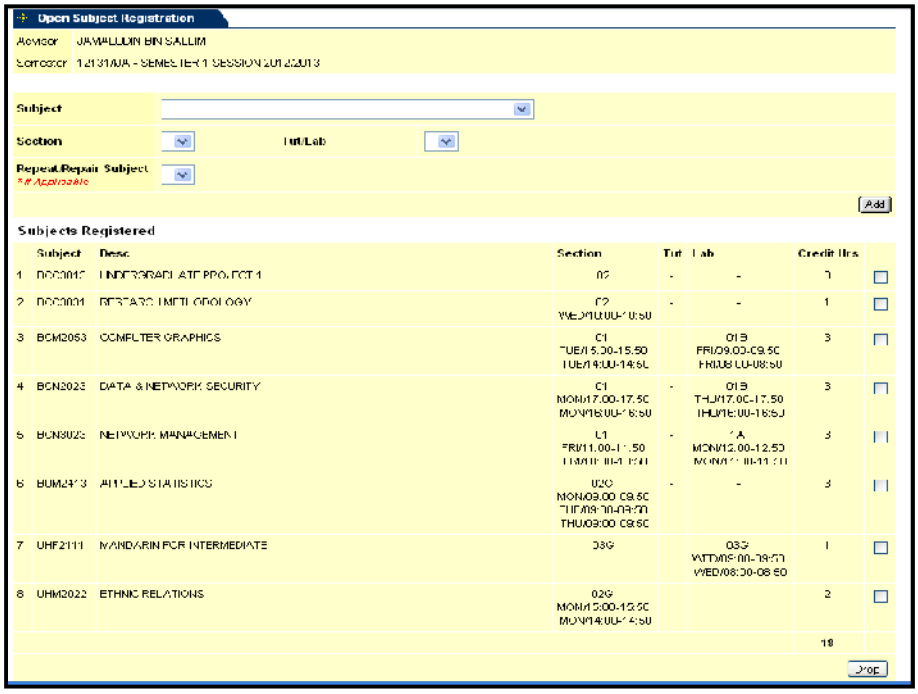

Figure 2.2.1 Overview of Open subject Registration part1

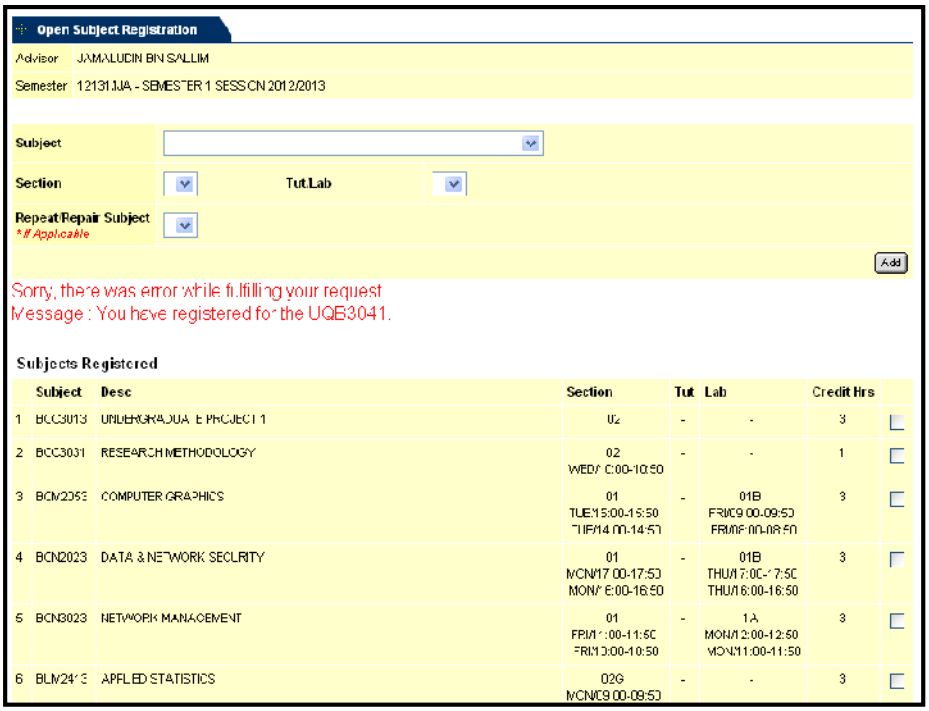

Figure 2.2.2 Overview of Open subject Registration part2

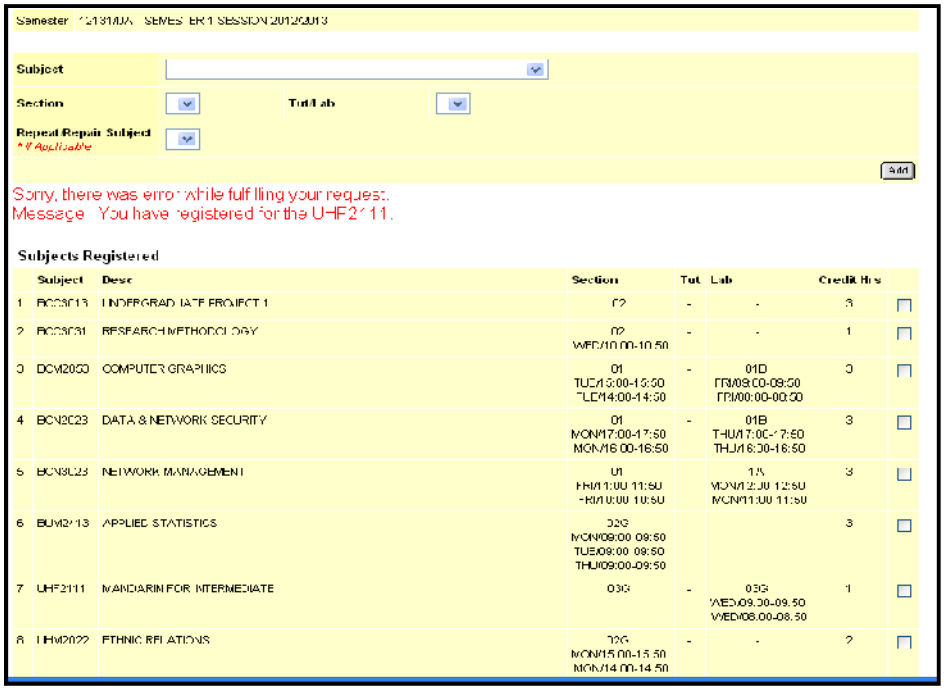

Figure 2.2.3 Overview of Open subject Registration part3

Hard constraints

- No student can be assigned to more than one course at the same time.
- The room should satisfy the features required by the courses.
- The number of students attending the course should be less than or equal to the capacity of the room.
- No more than one course is allowed at a timeslot in each room.

Soft constraints

- A student should not attend only one course in a day.
- A student should not attend more than two courses consecutively.
- A student should not attend a course in the last period in any day.

Figure 2.2.4 Example soft constraints and hard constraints [9]

In general, the set of constraint can be categorized as hard and soft[18]. Hard constraints are those that are compulsory to be fulfilled. A timetable will not be acceptable if any of the hard constraint is violated. Soft constraints include some non-compulsory requirements. Soft constraints might be violated but the number of violations had to be minimized in order to increase the quality of the timetable. A timetable without any hard constraints violations will be referred to as a feasible timetable[18] in which all courses are assigned to periods and rooms and satisfy all hard constraints[9].

2.3 Heuristic Method

2.3.1 Technology using Heuristic Method

Define a heuristic function, h(n), that estimates the "goodness" of a node n. Specifically, h(n) = estimated cost or distance of minimal cost path from n to a goal state[15]. The heuristic function is an estimate, based on domain-specific information that is computable from the current state description. All domain knowledge used in the search is encoded in the heuristic function h.

2.3.2 How does Heuristic Method Work

Basically, the main technologies used to implement the method If h1 (n) < h2 (n) <= h*(n) for all n, h2 is better than dominates h1. Relaxing the problem such as remove constraints to create a much easier problem, use the solution cost for this problem as the heuristic function [16]. Combining heuristics will take the max of several admissible heuristics and still have an admissible heuristic, and it's better to use statistical estimates to compute lose admissibility. The following diagram shows an overview of heuristic method which is the reference from [17].

Algorithm 1: Pseudo code of the heuristic method DT-ET introduced in Sridharan and Zhou (1996)

Input: $\alpha, \beta, d, n \in \mathbb{Z}_+ p, r \in \mathbb{Z}_+^n$.
Output: Sequence π, completion time $C \in \mathbb{Z}^n$ and objective function value $f \in \mathbb{Z}$.
DT-ET($\alpha, \beta, d, n, p, r$)

```
1  begin
2      f ← 0
3      for i ← 1 to n do
4          Set current time t_0
5          Build set S(t_0) of candidate jobs to be sequenced at position i
6          foreach j ∈ S(t_0) do
7              Consider sequencing job j at position i and find a feasible completion time Ĉ_j
8              Estimate the total cost κ_j associated with this choice, considering jobs in S(t_0) \ {j}
9          Compute j_min = arg min_{j∈S(t_0)} {κ_j}
10         π_i ← j_min and C_i ← Ĉ_{j_min}
11         f ← f + α [d − C_i]_+ + β [C_i − d]_+
12     return π, C and f
13 end
```

Figure 2.3.2 Overview of Heuristic Method

2.4 Existing Techniques

2.4.1 Data Flow Diagram

The Data Flow Model consists of a set of integrated Data Flow Diagrams (DFD) which supported by appropriate documentation. DFD will represents the processes (Example: student registration process), data stores (Example: The information will be kept in Access as database), external entities (Example: Student, exam subject, staff) and data flows. Thus, by implementing course registration system, we can able to visualize how this system works. There are some advantages using this data flow modelling as one of the technique. Data flow modelling is a simple graphical technique which is easy to understand. Other than that, data flow modelling helps in determining whether the data store is appropriate for the generated output information as mentioned in [14]. Data flow modelling aids in my project by explaining the logic behind the data flow within the system.

2.4.2 Database Life Cycle

The DBLC is the cycle of development and changes that relational database goes through during the course of its life [6]. The cycle typically consists of several stages. This represents an admission that a full understanding of a problem and its solution is likely to evolve as the various stages of design and implementation proceed. The following diagram shows an overview of DBLC which is the reference from [7].

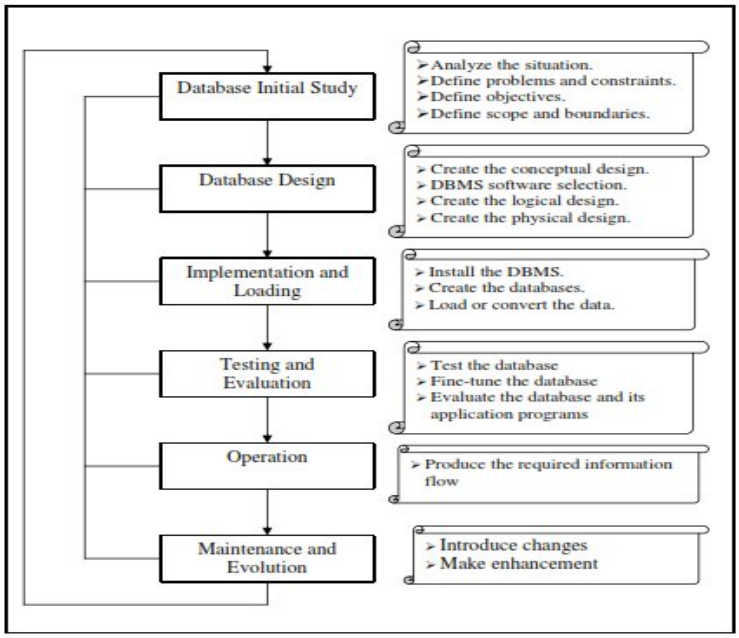

Figure 2.4.2 Overview of Database Life Cycle (DBLC)

2.5 Previous method to solve timetabling problem

2.5.1 Technology using Hill Climbing

Hill climbing is an optimization technique which belongs to the family of local search which was used for the timetabling problem by Appleby et al. (1960) [11]. It is relatively simple to implement and making it a popular first choice the relative simplicity of the algorithm makes it a popular first choice amongst optimizing algorithms. It is used widely in artificial intelligence, for reaching a goal state from a starting node. Choice of next node and starting node can be

varied to give a list of related algorithms. Although more advanced algorithms such as simulated annealing or tabu search may give better results, in some situations hill climbing works just as well. Hill climbing can often produce a better result than other algorithms when the amount of time available to perform a search is limited, such as with real-time systems. It is an anytime algorithm: it can return a valid solution even if it's interrupted at any time before it ends. The following diagram shows an overview of Hill climbing which is the reference from [9].

```
function HILL-CLIMBING( problem) returns a state that is a local maximum
   inputs: problem, a problem
   local variables: current, a node
                    neighbor, a node
   current ← MAKE-NODE(INITIAL-STATE[problem])
   loop do
      neighbor ← a highest-valued successor of current
      if VALUE[neighbor] ≤ VALUE[current] then return STATE[current]
      current ← neighbor
   end
```

Figure 2.5.1 Overview of Hill Climbing

2.5.2 Technology using Tabu Search (TS)

Tabu search is one of the local search techniques that are popular and applied to a lot of aspects of optimisations problems. According to Qu et al (2009) [20], the basic procedure of this technique can be divided into two main phases: intensification phase and diversification phase. This technique works by utilizing its components such as *type of tabu list, aspiration criteria, neighbourhoo*d strategies and etc, with intention to search the best improved solutions. Tabu search uses a local or neighbourhood search procedure to iteratively move from one potential solution to an improved solution in the neighbourhood of, until some stopping criterion has been satisfied generally, an attempt limit or a score threshold. The following diagram shows an overview of Tabu Search (TS) which is the reference from [18].

Algorithm 2 Tabu search for the unweighted MAX-k-SAT problem

Require: X - initial truth assignment, ϕ - boolean formula over n variables in CNF form

Ensure: X - locally optimal solution with respect to the 1-flip neighborhood

```
improves ← true
while improves do
    run oblivious local search
    time_stamp[i] ← 0 ∀i
    improves ← false
    difference ← 0
    for time = 1 to n do
        if max{Δ(X, i) + difference} > 0 then
            improves ← true
            break
        else
            unsat = {i | variable i appears in an unsatisfied clause}
            earliest ← min{time_stamp[i] | i ∈ unsat}
            // recall: variables not in taboo list have time_stamp = 0
            j ← argmax_i{Δ(X, i) | i ∈ unsat and time_stamp[i] = earliest}
            difference ← difference + Δ(X, j)
            X[j] ← 1 − X[j]
            time_stamp[j] ← time
        end if
    end for
end while
```

Figure 2.5.2 Overview of Tabu Search (TS)

2.5.3 Technology using Simulated Annealing (SA)

Once of the most widely studied local search meta heuristics is Simulated Annealing. It was proposed as a general optimization technique by Kirkpatrick et al. (1983) [23] and has been repeatedly applied to solve a wide range of problems. Simulated annealing algorithm, a new point is randomly generated. The distance of the new point from the current point, or the extent of the search, is based on a probability distribution with a scale proportional to the temperature. The algorithm accepts all new points that lower the objective, but also, with a certain probability, points that raise the objective. By accepting points that raise the objective, the algorithm avoids being trapped in local minima in early iterations and is able to explore globally for better solutions. The following diagram shows an overview of Simulated Annealing (SA) which is the reference from [19].

```
function SIMULATED-ANNEALING( problem, schedule) returns a solution state
   inputs: problem, a problem
           schedule, a mapping from time to "temperature"
   local variables: current, a node
                    next, a node
                    T, a "temperature" controlling prob. of downward steps
   current ← MAKE-NODE(INITIAL-STATE[problem])
   for t ← 1 to ∞ do
       T ← schedule[t]
       if T = 0 then return current
       next ← a randomly selected successor of current
       ΔE ← VALUE[next] – VALUE[current]
       if ΔE > 0 then current ← next
       else current ← next only with probability e^(ΔE/T)
```

Figure 2.5.3 Overview of Simulated Annealing (SA)

2.5.4 Technology using Great Deluge Algorithm (GD)

Great Deluge algorithm is the method which accepts every solution whose objective function is less than or equal to the upper limit (level) B [8]. The value B is monotonically decreased during the search and bounds the feasible region of the search space. In a typical implementation of the GD, the algorithm starts with a poor approximation, S, of the optimum solution. A numerical value called the badness is computed based on S and it measures how undesirable the initial approximation is. The higher the value of badness the more undesirable is the approximate solution. Another numerical value called the tolerance is calculated based on a number of factors, often including the initial badness. The following diagram shows an overview of Great Deluge Algorithm (GD) which is the reference from [20].

```
Great Deluge Algorithm
    Set estimated quality of every solution, EstimatedQuality = f(Sol_i) – F_i, where
    (i=1,…,population size) and F is a total force taken from Figure 3;
    Calculate force decay rate, β = EstimatedQuality/NumOfIte_GD;
    Set iteration_GD ← 0;
    for (iteration_GD< NumOfIte_GD)
        Define a randomly selected neighbourhood structure (Nbs1 or Nbs2) on Sol
        to generate a new solution called Sol*;
        Calculate f(Sol*);
        if (f(Sol*) < f(Sol_best))
            Sol ← Sol*;
            Sol_best ← Sol*;
        else
            if (f(Sol*)≤ level)
                Sol ← Sol*;
            endif
        endif
        level = level - β;
        Increase iteration_GD by 1;
    end for
```

Figure 2.5.4 Overview of Great Deluge Algorithm (GD)

After analyze all four method, the result is there are some minus in every method. Even though there are some major problems but the overcome to the attendance process can be done very effectively. Thus as a solution, best feature are sort list to be include in heuristic method. One challenge to the research community is therefore to explore how new search methodologies can underpin the development of more widely applicable timetabling systems. Indeed this is one of the main motivating factors for the current level of interest in hyper heuristic research.

CHAPTER 3

METHODOLOGY

This chapter is the main part of this system because it has the design, prototype, research and related software and hardware for this system. The course registration system is developed by using the concept of System Development Life Cycle (SDLC). SDLC is the process use to create or altering information system and the models and methodologies that wil used to develop the system. Besides that, this chapter includes a detailed study of the business needs of the organization with problem analysis process with performing data flow diagram. In enhance to that the software and hardware requirement has been discovered.

3.1 System Development Life Cycle (SDLC)

An SDLC (System Development Life Cycle) has three primary business objectives:

- Ensure the delivery of high quality systems;
- Provide strong management controls;
- Maximize productivity.

In other words, the SDLC ensure to produce more function, with higher quality, in less time, with less resource and in a predictable manner.

Figure 3.1.1 System Development Life Cycles

3.1.1 The Justification Choosing System Development Life Cycle

The Systems Development Life Cycle (SDLC) is a methodology that is used to describe process for building information systems. SDLC is one of the best way to develop a system as it result in a high quality system that meets or exceeds customer expectations, reaches completion within time and cost estimates, works effectively and is inexpensive to maintain and cost-effective to enhance as mentioned in [1]. SDLC also often used because of it can manage the level of complexity. Besides that, there are quite number of SDLC models which are waterfall, spiral, Agile software development rapid prototyping, incremental and synchronize and stabilize as it is stated in [2].

Furthermore, SDLC is also intended to develop information in a very purposeful, structured and methodical way, repeating each stage of the life cycle. The SDLC originated in the 1960's to develop large scale functional business systems in an age of large scale business conglomerates. Information systems activities revolved around heavy data processing and number crunching routines as it is mentioned in [1].